AMENDMENTS TO THE CLAIMS:

Please amend the claims as follows, substituting any amended claim(s) for the corresponding pending claim(s):

1. (Currently Amended) For use in a redundant, high-availability system of processor-based components, a system for memory equalization comprising:

a memory containing data elements each stored within one of a plurality of defined memory segments, wherein the data elements within the memory segments may be selectively changed;

a direct memory access circuit ~~capable of~~ automatically copying memory segments from the memory to a queue without utilizing a processor changing the memory segments;

a data link coupled to the queue, wherein each of the plurality of memory segments is structured to form a data packet which may be transmitted without internal changes over the data link; and

a control preventing the memory segments from receiving changes to the data elements contained therein ~~at a rate~~ faster than ~~a~~ transfer ~~rate~~ of data elements within the memory segments over the data link.

2. (Original) The system as set forth in Claim 1, further comprising:

a processor intermittently changing data elements within the memory segments, wherein the data link operates without direct control by the processor.

3. (Original) The system as set forth in Claim 1, wherein the memory further comprises a set of memory segments mapping to a corresponding set of memory segments within a device coupled to the data link.

4. (Original) The system as set forth in Claim 1, wherein the memory further comprises a portion of Firewire Global Memory and wherein the data link comprises a Firewire data link.

5. (Original) The system as set forth in Claim 1, wherein each of the data elements further comprise one of the following:

call state information for a call being processed;

resource allocation records for resources allocated to a call being processed; and

other information regarding a call being processed.

6. (Original) The system as set forth in Claim 1, wherein the direct memory access circuit copies memory segments containing changed data elements therein from the memory to the queue, the copying of a memory segment being triggered by a processor which intermittently changes data elements within the memory segments.

7. (Original) The system as set forth in Claim 6, wherein the processor writes a direct memory access descriptor for a memory segment containing a changed data element and sets controls bits initiating copying of the memory segment by the direct memory access circuit.

8. (Original) The system as set forth in Claim 1, wherein the direct memory access circuit copies memory segments containing changed data elements therein from the memory to the queue, the copying of a memory segment being triggered by a circuit monitoring writes to preselected memory addresses to detect changes to a data element within any memory segment.

9. (Original) The system as set forth in Claim 1, wherein the direct memory access circuit sequentially copies each of the memory segments from the memory to the queue in a continuous loop.

10. (Currently Amended) The system as set forth in Claim 1, wherein the data link ~~comprises~~ operates in accordance with an asynchronous transfer mode ~~switch~~.

11. (Currently Amended) A redundant, high-availability system of processor-based components, comprising:

an active component;

a standby component; and

a system for memory equalization between the active and standby components comprising:

counterpart memories within the active and standby components each containing data elements stored within one of a plurality of defined memory segments mapped to addresses within both of the counterpart memories, wherein the data elements within the memory segments may be selectively changed;

a direct memory access circuit within each of the active and standby components, the direct memory access circuit within the active component capable of automatically copying memory segments from the memory within the active component to a queue within the active component without utilizing a processor within the active component changing the memory segments within the active component and the direct memory access circuit within the standby component capable of automatically copying memory segments from a queue within the standby component to the memory within the standby component without utilizing a processor within the standby component changing the memory segments within the standby component;

a data link coupling to the queue within the active component to the queue within the standby component, wherein each of the plurality of memory segments is structured to form a data packet which may be transmitted without internal changes over the data link; and

a control within the active component preventing the memory segments within the memory in the active component from receiving changes to the data elements contained therein ~~at a rate~~ faster than ~~a~~ transfer ~~rate~~ of data elements within the memory segments over the data link.

12. (Original) The system as set forth in Claim 11, wherein the active component further comprises:

a processor intermittently changing data elements within the memory segments, wherein the data link operates without direct control by the processor.

13. (Original) The system as set forth in Claim 11, wherein the counterpart memories within the active and standby components each further comprise a set of memory segments mapping to a common set of addresses.

14. (Original) The system as set forth in Claim 11, wherein the counterpart memories within the active and standby components further comprise a portion of Firewire Global Memory and wherein the data link comprises a Firewire data link.

15. (Original) The system as set forth in Claim 11, wherein each of the data elements within either of the counterpart memories further comprise one of the following:

call state information for a call being processed;

resource allocation records for resources allocated to a call being processed; and

other information regarding a call being processed.

16. (Original) The system as set forth in Claim 11, wherein the direct memory access circuit within the active component copies memory segments containing changed data elements therein from the memory within the active component to the queue within the active component, the copying of a memory segment being triggered by a processor within the active component which intermittently changes data elements within the memory segments within the active component.

17. (Original) The system as set forth in Claim 16, wherein the processor within the active component writes a direct memory access descriptor for a memory segment within the active component which contains a changed data element and sets controls bits initiating copying of the memory segment within the active component by the direct memory access circuit within the active component.

18. (Original) The system as set forth in Claim 11, wherein the direct memory access circuit within the active component copies memory segments containing changed data elements therein from the memory within the active component to the queue within the active component, the copying of a memory segment being triggered by a circuit within the active component which monitor writes to preselected memory addresses to detect changes to a data element within any memory segment within the active component.

19. (Original) The system as set forth in Claim 11, wherein the direct memory access circuit within the active component sequentially copies each of the memory segments from the memory within the active component to the queue within the active component in a continuous loop.

20. (Original) The system as set forth in Claim 11, wherein a controller for the data link within the standby component verifies data integrity for data packets received over the data link and acknowledges successful transfer of data packets over the data link to the active component.

21. (Original) The system as set forth in Claim 11, wherein the direct memory access circuit within the standby component automatically moves memory segments from a queue within the standby component coupled to the data link into a corresponding memory location in the memory within the standby component.

22. (Currently Amended) The system as set forth in Claim 11, wherein the data link ~~comprises~~ operates in accordance with an asynchronous transfer mode ~~switch~~.

23. (Original) The system as set forth in Claim 11, wherein the active component is one of N active components supported by the standby component.

24. (Currently Amended) For use in a redundant, high-availability system of processor-based components, a method of memory equalization comprising the steps of:

selectively changing data elements each stored within one of a plurality of defined memory segments contained within a memory;

automatically copying memory segments from the memory to a queue utilizing a direct memory access circuit and without utilizing a processor changing the memory segments;

transferring memory segments over a data link coupled to the queue, wherein each of the plurality of memory segments is structured to form a data packet which may be transmitted without internal changes over the data link; and

inhibiting the memory segments from receiving changes to the data elements contained therein ~~at a rate~~ faster than ~~a~~ transfer ~~rate~~ of data elements within the memory segments over the data link.

25. (Original) The method as set forth in Claim 24, further comprising:

intermittently changing data elements within the memory segments utilizing a processor, wherein the data link operates without direct control by the processor.

26. (Original) The method as set forth in Claim 24, further comprising:

mapping the memory segments to a common set of addresses for counterpart memory segments within a standby component.

27. (Original) The method as set forth in Claim 26, wherein the steps of mapping the memory segments to a common set of addresses for counterpart memory segments within a standby component and transferring memory segments over a data link coupled to the queue further comprise:

mapping a portion of Firewire Global Memory to the memory segments and the counterpart memory segments; and

transferring memory segments over a Firewire data link.

28. (Original) The method as set forth in Claim 24, further comprising:

storing one of the following within each of the data elements:

call state information for a call being processed;

resource allocation records for resources allocated to a call being processed; and

other information regarding a call being processed.

29. (Original) The method as set forth in Claim 24, wherein the step of automatically copying memory segments from the memory to a queue utilizing a direct memory access circuit further comprises:

initiating copying of a memory segment by a processor within an active component containing the memory segment, wherein the processor intermittently changes data elements within the memory segments within the active component.

30. (Original) The method as set forth in Claim 29, further comprising:

writing a direct memory access descriptor for a memory segment within the active component which contains a changed data element; and

setting controls bits initiating copying of the memory segment which contains the changed data element by the direct memory access circuit.

31. (Original) The method as set forth in Claim 24, wherein the step of automatically copying memory segments from the memory to a queue utilizing a direct memory access circuit further comprises:

initiating copying of a memory segment by a circuit within an active component containing the memory segment, wherein the circuit monitor writes to preselected memory addresses to detect changes to a data element within any memory segment within the active component.

32. (Original) The method as set forth in Claim 24, wherein the step of automatically copying memory segments from the memory to a queue utilizing a direct memory access circuit further comprises:

utilizing the direct memory access circuit, sequentially copying each memory segment from the memory to the queue in a continuous loop.

33. (Original) The method as set forth in Claim 24, wherein the step of transferring memory segments over a data link coupled to the queue further comprises:

transmitting the memory segments to a standby component coupled to the data link, wherein a controller for the data link within the standby component verifies data integrity for data packets received over the data link and acknowledges successful transfer of data packets over the data link to an active component containing the queue.

34. (Original) The method as set forth in Claim 33, wherein the step of transmitting the memory segments to a standby component coupled to the data link further comprises:

receiving the memory segments in a queue within the standby component, wherein a direct memory access circuit within the standby component automatically moves memory segments from the queue within the standby component into a corresponding memory location in a memory within the standby component.

35. (Original) The method as set forth in Claim 24, wherein the step of transferring memory segments over a data link coupled to the queue further comprises:

transmitting the memory segments utilizing an asynchronous transfer mode switch.

36. (Original) The method as set forth in Claim 24, further comprising:

supporting processing within an active component containing the memory and the queue utilizing a standby component coupled to the data link, wherein the active component is one of N active components supported by the standby component.